# Refine Search

**Search Results -**

| Terms | Documents |
|---|---|
| Loop And (do ADJ while) or (do-while) and pipelining | 10 |

**Database:**
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

**Search:** L3

Refine Search | Recall Text | Clear | Interrupt

## Search History

**DATE: Sunday, November 28, 2004** Printable Copy Create Case

| Set Name side by side | Query | Hit Count | Set Name result set |
|---|---|---|---|
| *DB=USPT; PLUR=NO; OP=OR* | | | |
| L3 | Loop And (do ADJ while) or (do-while) and pipelining | 10 | L3 |
| L2 | L1 AND loop | 36 | L2 |
| L1 | ingberg.xp. | 107 | L1 |

END OF SEARCH HISTORY

# Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

**Search Results** - Record(s) 1 through 36 of 36 returned.

☐ 1. Document ID: US 6820258 B1

L2: Entry 1 of 36 File: USPT Nov 16, 2004

US-PAT-NO: 6820258
DOCUMENT-IDENTIFIER: US 6820258 B1

TITLE: System and method for dynamically optimizing executing activations

DATE-ISSUED: November 16, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Fink; Stephen J. | Yorktown Heights | NY | | |
| Wegman; Mark | Ossining | NY | | |

US-CL-CURRENT: 717/158

ABSTRACT:

A system and method for dynamically optimizing a code sequence of a program while executing in a computer system comprises: identifying one or more program yield points in an original code sequence at which a run-time representation of the original code sequence may be optimized in an executing program; generating a prologue of instructions for setting up program state associated with the original code sequence at a particular yield point; adding the prologue of instructions to an intermediate representation of the original code sequence code for generating a specialized code sequence; and, compiling the specialized code sequence with a compiler for generating a run-time representation of the specialized code sequence, the run-time representation being further optimized for execution on a target computer system.

20 Claims, 3 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 3

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

☑ 2. Document ID: US 6820250 B2

L2: Entry 2 of 36 File: USPT Nov 16, 2004

US-PAT-NO: 6820250

DOCUMENT-IDENTIFIER: US 6820250 B2

TITLE: Mechanism for software pipelining loop nests

DATE-ISSUED: November 16, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Muthukumar; Kalyan | Cupertino | CA | | |
| Doshi; Gautam B. | Sunnyvale | CA | | |

US-CL-CURRENT: 717/116; 717/150

ABSTRACT:

A method is provided for processing nested loops that include a modulo-scheduled inner loop within an outer loop. The nested loop is scheduled to execute the epilog stage of the inner loop for a given iteration of the outer loop with the prolog stage of the inner loop for the next iteration of the outer loop. For one embodiment of the invention, this is accomplished by initializing an epilog counter for the inner loop to a value that bypasses draining the software pipeline. This causes the processor to exit the inner loop before it begins draining the inner loop pipeline. The inner loop pipeline is drained during the next iteration of the outer loop, while the inner loop pipeline fills for the next iteration of the outer loop.

29 Claims, 11 Drawing figures
Exemplary Claim Number: 9
Number of Drawing Sheets: 10



---

☐ 3. Document ID: US 6810519 B1

L2: Entry 3 of 36 File: USPT Oct 26, 2004

US-PAT-NO: 6810519
DOCUMENT-IDENTIFIER: US 6810519 B1

TITLE: Achieving tight binding for dynamically loaded software modules via intermodule copying

DATE-ISSUED: October 26, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Hicks; Daniel Rodman | Byron | MN | | |

US-CL-CURRENT: 717/166

ABSTRACT:

A computer system, a computer product and a method in which static storage within an environment comprising a plurality of compilation modules is managed such that compiled cloned copies of called externally resolved (with respect to a compilation unit) items are preferentially executed in favor of the corresponding externally resolved item based on a favorable comparison of version information of version information prior to execution. In one embodiment, JAVA.RTM. programming language methods are processed within the context a modified framework.

7 Claims, 23 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 15



---

☐ 4. Document ID: US 6785882 B1

L2: Entry 4 of 36 File: USPT Aug 31, 2004

US-PAT-NO: 6785882
DOCUMENT-IDENTIFIER: US 6785882 B1

TITLE: Process-driven tool interface for an object management system

DATE-ISSUED: August 31, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Goiffon; David A. | Shoreview | MN | | |
| Hartmann; Gerald E. | Minneapolis | MN | | |
| Johnson; David R. | Oakdale | MN | | |

US-CL-CURRENT: 717/120; 707/200

ABSTRACT:

A process-driven object management system for managing data and code modules is disclosed. The object management system includes a repository that stores objects, wherein ones of the objects referred to as "Asset elements" each describe a respective code or data module. The object management system includes a set of scripted tools for performing renovation, transformation, and code development tasks on the code and data modules. According to one aspect of the invention, the tool invocation constructs are stored as objects in the repository, such that some of the same object management tools and automated repository interface functions used to manage the Asset element can also be used to manage and veiw the tool objects.

46 Claims, 20 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 17

☐ 5. Document ID: US 6782374 B2

L2: Entry 5 of 36 File: USPT Aug 24, 2004

US-PAT-NO: 6782374
DOCUMENT-IDENTIFIER: US 6782374 B2

TITLE: System, method and article of manufacturing for a runtime program analysis tool for a simulation engine

DATE-ISSUED: August 24, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Nichols; Mark Stewart | Downers Grove | IL | | |

US-CL-CURRENT: 706/45; 706/15

ABSTRACT:

A system is disclosed that provides a goal based learning system utilizing a rule based expert training system to provide a cognitive educational experience. The system provides the user with a simulated environment that presents a business opportunity to understand and solve optimally. Mistakes are noted and remedial educational material presented dynamically to build the necessary skills that a user requires for success in the business endeavor. The system utilizes an artificial intelligence engine driving individualized and dynamic feedback with synchronized video and graphics used to simulate real-world environment and interactions. Multiple "correct" answers are integrated into the learning system to allow individualized learning experiences in which navigation through the system is at a pace controlled by the learner. A robust business model provides support for realistic activities and allows a user to experience real world consequences for their actions and decisions and entails realtime decision-making and synthesis of the educational material. The system includes tools for analysis and display of a presentation as it is presented.

19 Claims, 79 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 58

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

☐ 6. Document ID: US 6772415 B1

L2: Entry 6 of 36 File: USPT Aug 3, 2004

US-PAT-NO: 6772415
DOCUMENT-IDENTIFIER: US 6772415 B1

TITLE: Loop optimization with mapping code on an architecture

DATE-ISSUED: August 3, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Danckaert; Koen | Asse | | | BE |
| Catthoor; Francky | Temse | | | BE |

US-CL-CURRENT: 717/161; 712/241

ABSTRACT:

A loop transformation step, to be performed on code and improving data transfer and storage, while executing said transformed code on a parallel processor, is disclosed. Improval of the data locality and regularity of the algorithm, described by said code, is aimed at. Said loop transformation step works globally and is feasible for realistic code sizes.

47 Claims, 45 Drawing figures
Exemplary Claim Number: 47
Number of Drawing Sheets: 28



---

☐ 7. Document ID: US 6769113 B1

L2: Entry 7 of 36 File: USPT Jul 27, 2004

US-PAT-NO: 6769113
DOCUMENT-IDENTIFIER: US 6769113 B1

TITLE: Enterprise process models and enterprise application for information technologies

DATE-ISSUED: July 27, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Bloom; Bard | Yorktown Heights | NY | | |
| Reimer; Darrell | Ossining | NY | | |
| Simmonds; Ian D. | Dobbs Ferry | NY | | |
| Wegman; Mark N. | Ossining | NY | | |

US-CL-CURRENT: 717/103; 705/8

ABSTRACT:

An enterprise process model that comprises a plurality of actors, actions performed by said actors, objects acted upon by said actions, and roles. Data characterizing the view of the attributes of at least one of the objects of the model is generated by associating a plurality of situation/role pairs with the at least one object; and then, for each particular situation/role pair, defining a view definition for subsequent use. In another aspect of the present invention, an enterprise application for use in an information system that comprises diverse software services and hardware platforms is generated by providing a model of the enterprise

process, wherein the model comprises a plurality of actors, actions performed by said actors, objects acted upon by said actions, and roles. A configuration defining software services and hardware platforms that support the model is generated. Finally, program fragments that support the model are generated.

12 Claims, 9 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 9



---

☐ 8. Document ID: US 6763327 B1

L2: Entry 8 of 36 File: USPT Jul 13, 2004

US-PAT-NO: 6763327
DOCUMENT-IDENTIFIER: US 6763327 B1

TITLE: Abstraction of configurable processor functionality for operating systems portability

DATE-ISSUED: July 13, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Songer; Christopher Mark | Mountain View | CA | | |
| Konas; Pavlos | Mountain View | CA | | |
| Gauthier; Marc E. | Sunnyvale | CA | | |
| Chea; Kevin C. | San Francisco | CA | | |

US-CL-CURRENT: 703/21; 703/20, 703/22, 703/23, 703/27, 716/1, 716/17, 716/18, 716/2, 717/107, 717/138, 717/151, 717/163 , 718/1, 719/310, 719/319, 719/321, 719/324, 719/328

ABSTRACT:

A hardware abstraction layer operates as a system architectural layer between a real-time operating system and an underlying configurable processor. The hardware abstraction layer provides an abstraction of processor-specific functionality to the operating system. In particular, it abstracts configurable processor features visible to the operating system to provide a uniform, standardized interface between the operating system and the configurable processor on which it runs. Thus, an operating system running on top of the hardware abstraction layer will work on all configurations of the processor which differ from one another only in the configuration parameters covered by the hardware abstraction layer. The hardware abstraction layer may be generated using the same information that is used to describe the features being configured in the configurable processor. Automatic generation of the HAL greatly eases use of the HAL and the configurable processor, since the user is not required to manually write the HAL or adapt an existing one based on the processor configuration parameters.

16 Claims, 3 Drawing figures
Exemplary Claim Number: 1

Number of Drawing Sheets: 2

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

---

☐ 9. Document ID: US 6760903 B1

L2: Entry 9 of 36 File: USPT Jul 6, 2004

US-PAT-NO: 6760903
DOCUMENT-IDENTIFIER: US 6760903 B1

TITLE: Coordinated application monitoring in a distributed computing environment

DATE-ISSUED: July 6, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Morshed; Farokh | Amherst | NH | | |
| Meagher; Robert | Milford | NM | | |

US-CL-CURRENT: 717/130

ABSTRACT:

Techniques for gathering execution information about an application, such as a distributed application, are described. Key communication points in cross execution context calls, such as remote procedure calls, are determined and control is transferred to interception routines to insert and extract execution information. Outgoing remote procedure calls are intercepted on a client that inserts call origin information into the request sent to a server system. The server system intercepts and extracts the call origin information and additionally inserts other information in a response sent to the client system upon completion of a remote procedure call. In turn, the client system intercepts the response and extracts other performance information. On each client and server system, information is gathered by a reader and forwarded to a local collector. Program execution data may be collected and correlated for coordinated application monitoring.

50 Claims, 82 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 77

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

---

☐ 10. Document ID: US 6757888 B1

L2: Entry 10 of 36 File: USPT Jun 29, 2004

US-PAT-NO: 6757888
DOCUMENT-IDENTIFIER: US 6757888 B1

TITLE: Method and apparatus for manipulating data during automated data processing

DATE-ISSUED: June 29, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Knutson; Loren G. | Allen | TX | | |
| Zurawski; John C. | Allen | TX | | |
| Simone, Jr.; Kenneth D. | Murphy | TX | | |
| Gharbia; Nezar A. | Richardson | TX | | |

US-CL-CURRENT: 717/109; 345/619, 382/276, 715/515, 715/523, 717/106

ABSTRACT:

A number of items of data from a data source (12) can be processed and then deposited in at least one data destination (16, 17). The data can be image data, text data, numeric data or some other type of data, or a combination of these types of data. Processing of the data is controlled by a project definition (14, 71, 101) which includes a plurality of modules selected from a variety of available modules (Tables 1-4). The modules have input and output ports that are interrelated by binding information. Some of the modules are capable of taking an item of data and splitting it into two or more component parts. Other modules are capable of taking separate items of data and combining them.

14 Claims, 19 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 10



---

☐ 11. Document ID: US 6754885 B1

L2: Entry 11 of 36 File: USPT Jun 22, 2004

US-PAT-NO: 6754885
DOCUMENT-IDENTIFIER: US 6754885 B1

TITLE: Methods and apparatus for controlling object appearance in a process control configuration system

DATE-ISSUED: June 22, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Dardinski; Steven | Westford | MA | | |
| Eldridge; Keith | North Easton | MA | | |
| Hall; Robert | South Easton | MA | | |
| Johnson; Mark | North Attleboro | MA | | |
| MacKay; Brian | Coppell | TX | | |
| Meskonis; Paul | Norwood | MA | | |
| Volk; Scott | North Easton | MA | | |

US-CL-CURRENT: 717/113

ABSTRACT:

The invention provides improved apparatus for configuring process, environmental, industrial and other control systems. Such apparatus employs "appearance" objects (or other data and/or programming constructs) defining the appearance of configurable system components in graphical editors or other views in which the components may be depicted. "Placeholder" objects (or other constructs) persist the location, size, color, or other aspects of appearance defined by an appearance object for a configurable component in views in which it is actually depicted. By way of example, a process control configuration apparatus according to this aspect of the invention uses "configurable" objects to define blocks, loops and other components of a process control system. Appearance objects provide (or reference) icons or representations indicating how the configurable objects are to be depicted, e.g., in a configuration editor. Placeholder objects are created for each configurable object that is placed in a configuration using that editor. The placeholder objects identify the sizes, locations, colors, etc., of the icons used in the editor to represent the configurable objects.

70 Claims, 121 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 75



---

☐ 12. Document ID: US 6748585 B2

L2: Entry 12 of 36 File: USPT Jun 8, 2004

US-PAT-NO: 6748585
DOCUMENT-IDENTIFIER: US 6748585 B2

TITLE: Computer programming language pronouns

DATE-ISSUED: June 8, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Proebsting; Todd A. | Redmond | WA | | |
| Zorn; Benjamin G. | Woodinville | WA | | |

US-CL-CURRENT: 717/136; 717/114, 717/116, 717/140, 717/141, 717/142, 717/143

ABSTRACT:

Programming language constructs called pronouns and referents, and a method, system, and apparatus for translating computer source code that contains the pronouns and referents. A referent is any semantic or syntactic construct in the source code (e.g., a statement, a portion of a statement, an expression, or a value) to which a pronoun refers. A pronoun is a programming-language defined source-code symbol or a sequence of symbols that refers to the referent. As a result, pronouns eliminate the need to define new names or macros for repeated program segments. When a translator encounters the pronoun in the source code, the

translator searches the source code for the referent and substitutes the referent for the pronoun. Thus, by using pronouns and referents, the programmer can write programs faster and easier and eliminate program redundancy without losing readability.

31 Claims, 22 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 13



---

☐ 13. Document ID: US 6732353 B1

L2: Entry 13 of 36 File: USPT May 4, 2004

US-PAT-NO: 6732353
DOCUMENT-IDENTIFIER: US 6732353 B1

TITLE: Method and system for generating enterprise applications of a diversity of information technologies

DATE-ISSUED: May 4, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Bloom; Bard | Yorktown Heights | NY | | |
| Reimer; Darrell | Ossining | NY | | |
| Simmonds; Ian D. | Dobbs Ferry | NY | | |
| Wegman; Mark N. | Ossining | NY | | |

US-CL-CURRENT: 717/103; 705/8

ABSTRACT:

An enterprise process model that comprises a plurality of actors, actions performed by said actors, objects acted upon by said actions, and roles. Data characterizing the view of the attributes of at least one of the objects of the model is generated by associating a plurality of situation/role pairs with the at least one object; and then, for each particular situation/role pair, defining a view definition for subsequent use.

In another aspect of the present invention, an enterprise application for use in an information system that comprises diverse software services and hardware platforms is generated by providing a model of the enterprise process, wherein the model comprises a plurality of actors, actions performed by said actors, objects acted upon by said actions, and roles. A configuration defining software services and hardware platforms that support the model is generated. Finally, program fragments that support the model are generated.

40 Claims, 9 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 9

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

☐ 14. Document ID: US 6725452 B1

L2: Entry 14 of 36 File: USPT Apr 20, 2004

US-PAT-NO: 6725452
DOCUMENT-IDENTIFIER: US 6725452 B1

TITLE: Method for resolving dependency conflicts among multiple operative entities within a computing environment

DATE-ISSUED: April 20, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Te'eni; Moddy | Tel Aviv | | | IL |
| Shufer; Ilan | Bet Dagan | | | IL |

US-CL-CURRENT: 717/168

ABSTRACT:

A computer-based method sets up automatically a computer system configuration in association with a computer system upgrade process. In accordance with the system configuration, the upgrade process installs a required set of operative elements in order to provide for the reliable operation of the computer system. The operatively correct combination of hardware, system and application software components is accomplished by inter-component dependency checking and conflict resolving procedures utilizing specific inter-component dependency rules tables and component definition and inter-component relationship definition databases.

17 Claims, 10 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 10

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

☐ 15. Document ID: US 6718533 B1

L2: Entry 15 of 36 File: USPT Apr 6, 2004

US-PAT-NO: 6718533
DOCUMENT-IDENTIFIER: US 6718533 B1

TITLE: Method for building a real-time control system with mode and logical rate

DATE-ISSUED: April 6, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Schneider; Stanley A. | Sunnyvale | CA | | |
| Chen; Vincent W. | San Jose | CA | | |
| Pardo-Castellote; Gerardo | Palo Alto | CA | | |
| Wang; Howard H. | Sunnyvale | CA | | |
| Joshi; Rajive | Sunnyvale | CA | | |

US-CL-CURRENT: 717/100; 700/97

ABSTRACT:

A development tool combines advantages of a simulation tool with an object-oriented modeling tool, including a real-time mathematical matrix library and an object model. The tool is applicable to any intelligent control system. A composite object group (COG) contains both sampled-data and event-driven capabilities. Mapping the computing resources of a computing device throughout the hierarchy of a control system is provided for modes, executable programs and threads of a processor. Within a given processor, each component of a control system may be assigned to individual execution threads within that processor simply by naming a thread.

23 Claims, 34 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 24



---

☑ 16. Document ID: US 6708331 B1

L2: Entry 16 of 36 File: USPT Mar 16, 2004

US-PAT-NO: 6708331
DOCUMENT-IDENTIFIER: US 6708331 B1

TITLE: Method for automatic parallelization of software

DATE-ISSUED: March 16, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Schwartz; Leon | Zichron Yaakov, 30900 | | | IL |

US-CL-CURRENT: 717/160

ABSTRACT:

The invention provides a scalable, automated, network friendly method for building parallel applications from embarrassingly parallel serial programs. Briefly, the steps of an exemplary method in this invention are as follows: First, the application loops with significant potential parallelism are identified. Second, from the set of loops identified, those loops which can statically be shown to not be parallelizable are disqualified. Next, the program is transformed into a parallel form in which the remaining identified loops are optimistically

parallelized and packaged into per-iteration functions. Then, shared memory in the functions is relocated to a shared memory section available to all computers and references to the shared memory in the source code are transformed into indirect accesses. Finally, the per-iteration functions are spawned on to multiple computers, at run-time, where each computer is given a range of iteration.

10 Claims, 12 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 6

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

☐ 17. Document ID: US 6684391 B1

L2: Entry 17 of 36 File: USPT Jan 27, 2004

US-PAT-NO: 6684391
DOCUMENT-IDENTIFIER: US 6684391 B1

TITLE: Method for operating a computer system, byte code verifier and computer system

DATE-ISSUED: January 27, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Stroetmann; Karl | Munich | | | DE |

US-CL-CURRENT: 717/148; 717/143

ABSTRACT:

The invention is directed to a method for operating a computer system, as well as to a byte code verifier and to a computer system. The inventive method checks whether a computer program loaded onto a computer system exercises an illegal access to a variable, i.e. whether the variable is initialized before it is read. This test ensues before the execution of the program, so that such a test no longer need be performed upon execution of the program. The inventive method requires little memory space in the testing of the program and nonetheless carries out a complete test. Since the testing ensues before the execution of the program, the program execution itself is considerably speeded up since no further test is thereby required.

11 Claims, 5 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

☐ 18. Document ID: US 6681383 B1

L2: Entry 18 of 36 File: USPT Jan 20, 2004

US-PAT-NO: 6681383
DOCUMENT-IDENTIFIER: US 6681383 B1

TITLE: Automatic software production system

DATE-ISSUED: January 20, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Pastor; Oscar | Valencia | | | ES |
| Iborra; Jose | Denia Alicante | | | ES |

US-CL-CURRENT: 717/126

ABSTRACT:

An automated software production system is provided, in which system requirements are captured, converted into a formal specification, and validated for correctness and completeness. In addition, a translator is provided to automatically generate a complete, robust software application based on the validated formal specification, including user-interface code and error handling code.

43 Claims, 7 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 7

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWMC | Draw. De

---

☐ 19. Document ID: US 6678886 B2

L2: Entry 19 of 36 File: USPT Jan 13, 2004

US-PAT-NO: 6678886
DOCUMENT-IDENTIFIER: US 6678886 B2

TITLE: Apparatus and method for generating optimization objects

DATE-ISSUED: January 13, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Kumon; Kouichi | Kawasaki | | | JP |

US-CL-CURRENT: 717/151; 717/135, 717/136, 717/140, 717/146, 717/154, 717/158, 717/159

ABSTRACT:

A system and method enable appropriately concentrating instruction strings or data pieces sporadically present in a plurality of regions over more than one

compilation unit and adjusting the front-and-rear relationship of executed instruction strings without changing the program compilation unit such as a file, subroutine, or function and also without creating a link processing program for batch processing of the system as a whole. Different section names are given to the executed instruction strings and the unexecuted instruction strings and the referenced data and the unreferenced data of an object program respectively. When an execution module is generated from the object program by linking, the sections having an executed section name and the sections having an unexecuted section name in a plurality of files may be aggregated respectively to divide the instructions into an execution portion and an unexecution portion.

25 Claims, 47 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 38



---

☐ 20. Document ID: US 6665866 B1

L2: Entry 20 of 36 File: USPT Dec 16, 2003

US-PAT-NO: 6665866
DOCUMENT-IDENTIFIER: US 6665866 B1
**** See image for Certificate of Correction ****

TITLE: Extensible compiler utilizing a plurality of question handlers

DATE-ISSUED: December 16, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Kwiatkowski; Paul | Seattle | WA | | |
| Richter; David | Seattle | WA | | |
| Aitken; William | Mercer Island | WA | | |
| Dickens; Brian | Bellevue | WA | | |
| Simonyi; Charles | Bellevue | WA | | |
| Paramasivam; Muthukrishnan | Seattle | WA | | |
| Eisner; Steve | Seattle | WA | | |
| Samaragdakis; Ioannis | Austin | TX | | |

US-CL-CURRENT: 717/159; 717/144, 717/158

ABSTRACT:

A compiler architecture uses a question and answer methodology between a reduction engine and nodes of a graph representing the program being compiled to provide for easy expandability of the compiler. By using the question and answer methodology, additional functionality can be added to the compiler by users, whether they be part of the original design team of the compiler or be a subsequent user.

51 Claims, 8 Drawing figures
Exemplary Claim Number: 1

Number of Drawing Sheets: 8

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

---

☐ 21. Document ID: US 6665865 B1

L2: Entry 21 of 36 File: USPT Dec 16, 2003

US-PAT-NO: 6665865
DOCUMENT-IDENTIFIER: US 6665865 B1

TITLE: Equivalence class based synchronization optimization

DATE-ISSUED: December 16, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Ruf; Erik | Redmond | WA | | |

US-CL-CURRENT: 717/157; 717/144, 717/151, 717/154, 717/156

ABSTRACT:

Synchronization optimization for statically compiled Java programs is performed in three phases: Thread closure analysis, Alias analysis, and Specialization and transformation. Thread closure analysis bounds the number of thread instances constructed at each thread allocation site, and determines the set of methods potentially executed by each thread instance. Alias analysis generates equivalence class representation based alias signatures for each method. These signatures describe the aliasing and synchronization behavior of each method. The specialization and transformation phase traverses a call graph in a top-down manner starting from the program entry point, and creates specialized copies of methods when they can be individually optimized. A synchronization operation is removed from the code whenever it can be proven that all objects reaching the operation at runtime can be synchronized by at most one thread instance.

30 Claims, 8 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 5

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

---

☐ 22. Document ID: US 6662357 B1

L2: Entry 22 of 36 File: USPT Dec 9, 2003

US-PAT-NO: 6662357
DOCUMENT-IDENTIFIER: US 6662357 B1
**** See image for Certificate of Correction ****

TITLE: Managing information in an integrated development architecture framework

DATE-ISSUED: December 9, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Bowman-Amuah; Michel K. | Colorado Springs | CO | | |

US-CL-CURRENT: 717/120

ABSTRACT:

A system, method, and article of manufacture are provided for managing information in a development architecture framework. Common information that is used by a plurality of components of a system is allowed to be accessed in a single, shared repository. Unique information that is unique to the components of the system is stored in corresponding designated folders. Media content communicated in the system is managed based on metadata thereof.

18 Claims, 14 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 14



---

☐ 23. Document ID: US 6662355 B1

L2: Entry 23 of 36 File: USPT Dec 9, 2003

US-PAT-NO: 6662355
DOCUMENT-IDENTIFIER: US 6662355 B1

TITLE: Method and system for specifying and implementing automation of business processes

DATE-ISSUED: December 9, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Caswell; Nathan S. | Yorktown Heights | NY | | |
| Ciccolo; Arthur C. | Ridgefield | CT | | |
| Nigam; Anil | Stamford | CT | | |

US-CL-CURRENT: 717/103; 705/1

ABSTRACT:

A method s for specifying and implementing automation of business processes where the specification is independently manipulable by both the business process owner and technical implementers, and resulting technical elements can be tested for compliance with every detail in the specification. The method creates a single shared model suitable for understanding and execution in both the business and technical domains by focusing on the specification problem in the area of business automation. The solution to the specification problem lies in Information,

Function, Flow (IFF or IF.sup.2) factorization of business processes. Models of the business are constructed by way of the IF.sup.2 modeling methodology. This is a complete model which includes, by construction, external specifications of each task included in the business model. The modularization problem is solved by preserving the partitioning of the system created in the business model. The automation system implements concrete modules that uniquely and directly correspond to particular elements whose external specification is determined by the business model.

9 Claims, 5 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 5

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

---

☐ 24. Document ID: US 6654951 B1

L2: Entry 24 of 36 File: USPT Nov 25, 2003

US-PAT-NO: 6654951
DOCUMENT-IDENTIFIER: US 6654951 B1

TITLE: Removal of unreachable methods in object-oriented applications based on program interface analysis

DATE-ISSUED: November 25, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Bacon; David Francis | New York | NY | | |
| Laffra; Johannes C. | Yorktown Heights | NY | | |
| Sweeney; Peter Francis | Spring Valley | NY | | |
| Tip; Frank | Mount Kisco | NY | | |

US-CL-CURRENT: 717/154; 717/151

ABSTRACT:

The present invention analyzes an application A and computes a set reachable methods in A by determining the methods in A that may be called from another reachable method in A, or from within a class library L used by A without analyzing the classes in L. The invention may be used as an optimization to reduce application size by eliminating unreachable methods.

19 Claims, 5 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 5

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

---

☐ 25. Document ID: US 6634026 B1

L2: Entry 25 of 36 File: USPT Oct 14, 2003

US-PAT-NO: 6634026
DOCUMENT-IDENTIFIER: US 6634026 B1

TITLE: Method and apparatus for correcting common errors in multiple versions of a computer program

DATE-ISSUED: October 14, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Jones; Robert Dennis | Hartland | WI | | |

US-CL-CURRENT: 717/170

ABSTRACT:

The present invention relates to a repair program for multiple versions of computer programs that have a common error by using a pattern search and substitution technique. The invention includes identifying a common error in a main computer program, finding a common code section that contains the common error, and locating a segment of the common code section that is modifiable. The code section is then modified by optimizing the code to perform the same functionality and adding additional code to correct the error. A repair program is then written to search other versions of the main computer program and perform the modification step automatically without having to manipulate the source or machine code manually on the various versions of the software.

20 Claims, 3 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 3


---

☐ 26. Document ID: US 6631518 B1

L2: Entry 26 of 36 File: USPT Oct 7, 2003

US-PAT-NO: 6631518
DOCUMENT-IDENTIFIER: US 6631518 B1
**** See image for Certificate of Correction ****

TITLE: Generating and utilizing organized profile information

DATE-ISSUED: October 7, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Bortnikov; Vita | Haifa | | | IL |

| | | | |
|---|---|---|---|
| Lambert; David John | Rochester | MN | |
| Mendelson; Bilha | Haifa | | IL |
| Roediger; Robert Ralph | Rochester | MN | |
| Schmidt; William Jon | Rochester | MN | |
| Shavit-Lottem; Inbal | Kibbutz Bet-Oren | | IL |

US-CL-CURRENT: 717/158

ABSTRACT:

Disclosed is a system and method for a profiling system wherein profile data is stored in a separable hierarchical fashion such that profile data for each compiled procedure in a computer program can be readily identified and utilized. In particular, each source module has a corresponding profile data file and each procedure has a corresponding procedure profile area. The system and method also includes a mechanism for verifying the existence and validity of profile information, and a mechanism for handling invalid profile information.

29 Claims, 12 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 8



---

☐ 27. Document ID: US 6618856 B2

L2: Entry 27 of 36 File: USPT Sep 9, 2003

US-PAT-NO: 6618856
DOCUMENT-IDENTIFIER: US 6618856 B2

TITLE: Simulation method and apparatus for use in enterprise controls

DATE-ISSUED: September 9, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Coburn; James D. | Cleveland Heights | OH | | |
| Hoskins; Josiah C. | Austin | TX | | |
| Brooks; Ruven E. | Shorewood | WI | | |

US-CL-CURRENT: 717/135; 700/86, 703/15, 703/22

ABSTRACT:

A method used with a simulator and a controller, the controller running execution code to provide output signals which, when linked to resources, cause the resources to cycle through requested activities, the simulator receiving controller output signals and, in response thereto, generating motion pictures of resources as the resources cycle through requested activities, the simulator using data structures which model the resources to determine which motion pictures to generate, the method for generating execution code and data structures for use by the controller

and the simulator, respectively, and comprising the steps of, for each resource, encapsulating resource information including resource logic in a control assembly (CA), instantiating at least one instance of at least one CA, compiling instantiated CA instance resource logic to generate execution code, gleaning simulation information from the instantiated CA instances and using the gleaned simulation information to generate a simulation data structure for the resources corresponding to the instantiated CA instances.

26 Claims, 129 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 103



---

☐ 28. Document ID: US 6611956 B1

L2: Entry 28 of 36 File: USPT Aug 26, 2003

US-PAT-NO: 6611956
DOCUMENT-IDENTIFIER: US 6611956 B1

TITLE: Instruction string optimization with estimation of basic block dependence relations where the first step is to remove self-dependent branching

DATE-ISSUED: August 26, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Ogawa; Hajime | Neyagawa | | | JP |
| Odani; Kensuke | Kyoto | | | JP |

US-CL-CURRENT: 717/152

ABSTRACT:

An instruction string optimization apparatus is provided which estimates the size of a constant to be resolved as an address difference before linking instructions. The apparatus comprises code dividing means (202) for dividing a serial assembler code (201) into basic blocks, size dependence relation generation means (204) for analyzing size dependence relations among the sizes of the instruction string between basic blocks, estimation order determining means (206) for determining the order of basic blocks in which the size of a constant to be resolved as an address difference is determined and size determining means (208) for determining the size of the constant in each basic block according to the determined order, whereby the size of a constant to be resolved as an address difference can be estimated to be a value close to and not less than its actual size, the number of codes can be reduced, and the process speed by a linker can be improved.

11 Claims, 63 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 37

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

☐ 29. Document ID: US 6609249 B2

L2: Entry 29 of 36 File: USPT Aug 19, 2003

US-PAT-NO: 6609249
DOCUMENT-IDENTIFIER: US 6609249 B2

TITLE: Determining maximum number of live registers by recording relevant events of the execution of a computer program

DATE-ISSUED: August 19, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Kunz; Robert C. | Palo Alto | CA | | |
| Dahl; Peter J. | Cupertino | CA | | |

US-CL-CURRENT: 717/161

ABSTRACT:

The present invention is a method and apparatus for compiler optimization that determines the maximum number of live computer registers, or pressure point. The present invention improves the productivity of a software developer by reducing compilation time of a computer program. More particularly, the overhead required during compilation to search information to determine the maximum number of live registers is reduced. The present invention records the relevant events related to the execution of a computer program, as opposed to a comprehensive history of the read instructions and write instructions. Also, the present invention maintains information about the maximum number of live registers for any partition related to the execution of a computer program. The present invention may bound the required system resources required to determine the maximum number of live registers to the number of registers associated with the number of partitions.

12 Claims, 11 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 11

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

☐ 30. Document ID: US 6601236 B1

L2: Entry 30 of 36 File: USPT Jul 29, 2003

US-PAT-NO: 6601236
DOCUMENT-IDENTIFIER: US 6601236 B1
** See image for Certificate of Correction **

TITLE: Cross platform program installation on drives using drive object

DATE-ISSUED: July 29, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Curtis; Bryce Allen | Round Rock | TX | | |

US-CL-CURRENT: 717/177

ABSTRACT:

Disclosed is a set of tools or program instructions, an installation program, and a system that operates a drive in a platform independent manner. A drive object represents a single drive mounted by an operating system and contains fields providing information including drive name, block size, free space, type, format long file name support, and space needed by the files to be installed on that drive. During the installation process, each drive is processed to ensure that the drive has sufficient free space for the files to be installed.

30 Claims, 4 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4



---

☐ 31. Document ID: US 6578192 B1

L2: Entry 31 of 36 File: USPT Jun 10, 2003

US-PAT-NO: 6578192
DOCUMENT-IDENTIFIER: US 6578192 B1

TITLE: Method and system for supporting dynamic document content expressed in a component-level language

DATE-ISSUED: June 10, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Boehme; Richard F. | Kent Lakes | NY | | |
| Duftler; Matthew J. | Tarrytown | NY | | |
| Epstein; David A. | Ossining | NY | | |
| Kesselman; Joseph J. | Ossining | NY | | |
| Weerawarana; Sanjiva | Yorktown Heights | NY | | |

US-CL-CURRENT: 717/115

ABSTRACT:

This invention provides a computer-method for parsing by enabling scripts to be expressed in a language which is syntax-compatible with the document surrounding them. A document is loaded having script and non-script components. Script components are identified and delineated, and are then passed to an interpreter,

which returns an object corresponding to each script component. Then, script elements in the original document are replaced with the last returned object from the interpreter. If the returned object is a suitable Document Object Model (DOM) Node, it replaces the script element in the document structure. If the object is not a DOM node, the server invokes its string conversion method to obtain a textual representation, and replaces the script element with that text. After all BML markup block elements in the document have been processed, the altered document is delivered to the client.

14 Claims, 4 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4



---

☑ 32. Document ID: US 6571385 B1

L2: Entry 32 of 36 File: USPT May 27, 2003

US-PAT-NO: 6571385
DOCUMENT-IDENTIFIER: US 6571385 B1
**** See image for Certificate of Correction ****

TITLE: Early exit transformations for software pipelining

DATE-ISSUED: May 27, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Muthukumar; Kalyan | Cupertino | CA | | |
| Chen; Dong-Yuan | Fremont | CA | | |
| Wu; Youfeng | Palo Alto | CA | | |
| Lavery; Daniel M. | Santa Clara | CA | | |

US-CL-CURRENT: 717/150; 712/219, 712/239

ABSTRACT:

The invention is directed to the transformation of software loops having early exit conditions, thereby allowing the loops to be more effectively converted to a single basic block for software pipelining. The invention assigns a predicate register for each early exit condition of the software loop. The predicate registers are set when the corresponding early exit condition is satisfied. In this manner, when the loop terminates the predicate registers can be examined to indicate which early exit conditions were satisfied. The invention produces loops having a lower recurrence II and resource II than conventional techniques.

18 Claims, 7 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 7

☐ 33. Document ID: US 6564179 B1

L2: Entry 33 of 36 File: USPT May 13, 2003

US-PAT-NO: 6564179
DOCUMENT-IDENTIFIER: US 6564179 B1

TITLE: DSP emulating a microcontroller

DATE-ISSUED: May 13, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Belhaj; Said O. | Coplay | PA | | |

US-CL-CURRENT: 703/26; 703/24, 703/25, 712/203, 712/209, 712/227, 712/28, 712/31, 712/34, 712/35, 712/36

ABSTRACT:

The present invention provides a processor device and technique having the capability of providing a two-processor solution with only one processor. In accordance with the principles of the present invention, a host processor is programmed in its native source and machine code language, and an emulated second processor is programmed in a different native source or machine code language particular to that emulated processor, to allow programming specialists in the different processors to develop common code for use on the same host processor. A multitasking operating system is included to allow time sharing operation between instructions from program code relating to the host processor (e.g., a DSP in the disclosed embodiment), and different program code relating to the emulated processor. The program code relating to the host processor (e.g., DSP) is written in program code which is native to the DSP, while the program code relating to the emulated processor (e.g., microcontroller) is written in program code which is native to the microcontroller. The SoftCore emulation module allows both DSP code and control code written for a microcontroller to execute independently on the same processor by multi-tasking resources in the faster, host processor (e.g., in the DSP), giving equal time slots of processor time to each processor (real and emulated).

4 Claims, 5 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 5



☐ 34. Document ID: US 6546362 B1

L2: Entry 34 of 36 File: USPT Apr 8, 2003

US-PAT-NO: 6546362
DOCUMENT-IDENTIFIER: US 6546362 B1

TITLE: Mold design system and recording medium

DATE-ISSUED: April 8, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Guo; Fu | Kawasaki | | | JP |
| Yoshikawa; Tadakatsu | Kawasaki | | | JP |

US-CL-CURRENT: 703/1; 703/2, 703/6, 703/7

ABSTRACT:

To quickly determine a parting line, a mold design system first obtains the orientations of faces constituting an article to be produced using a mold, and then classifies the faces according to their orientations. A boundary between faces that are classified into different sets is determined as a parting line. Thus, the parting line is automatically determined, permitting efficient mold design.

6 Claims, 22 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 22



---

☐ 35. Document ID: US 6539539 B1

L2: Entry 35 of 36 File: USPT Mar 25, 2003

US-PAT-NO: 6539539
DOCUMENT-IDENTIFIER: US 6539539 B1

TITLE: Active probes for ensuring software package compatibility

DATE-ISSUED: March 25, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Larsen; Mark S. | Hoffman Estates | IL | | |
| Liesen; Christopher D. | Naperville | IL | | |
| Zorn; Alan R. | Naperville | IL | | |

US-CL-CURRENT: 717/124; 379/9.01, 714/40, 717/121

ABSTRACT:

In a distributed computer program, active software probes in the form of small functions built into an application are invoked by another application. When invoked, an active probe provides a positive response if the service being requested is available from the probed package. If the service is not available, the probe will fail alerting the software package installer that there is a problem. The active probes thus perform a functionality check for the software

package, not a check merely based on the package's release number. Because the probe is active, it is capable of checking for subtending capabilities that the calling application may not realize are necessary for this service. This ensures full coverage of the test while hiding some implementation details from the calling application. By invoking these active probes upon installation of a new package, the installer can know immediately if the package has all services necessary to run correctly, thus reducing the possibility of a bad software package installation. The active software probes are thus capable of verifying software package compatibility at the time of program compilation as well as for program upgrades.

6 Claims, 6 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4



☐ 36. Document ID: US 6535861 B1

L2: Entry 36 of 36 File: USPT Mar 18, 2003

US-PAT-NO: 6535861
DOCUMENT-IDENTIFIER: US 6535861 B1

TITLE: Goal based educational system with support for dynamic characteristics tuning using a spread sheet object

DATE-ISSUED: March 18, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| O'Connor; Martha Torrey | Verona | NJ | | |
| Rosenfeld; Eren Tolga | New York | NY | | |

US-CL-CURRENT: 706/11; 434/107, 434/322, 434/327, 706/45, 706/46, 706/47, 706/60

ABSTRACT:

A system is disclosed that provides a goal based learning system utilizing a rule based expert training system to provide a cognitive educational experience. The system provides the user with a simulated environment that presents a business opportunity to understand and solve optimally. Mistakes are noted and remedial educational material presented dynamically to build the necessary skills that a user requires for success in the business endeavor. The system utilizes an artificial intelligence engine driving individualized and dynamic feedback with synchronized video and graphics used to simulate real-world environment and interactions. Multiple "correct" answers are integrated into the learning system to allow individualized learning experiences in which navigation through the system is at a pace controlled by the learner. A robust business model provides support for realistic activities and allows a student to experience real world consequences for their actions and decisions and entails realtime decision-making and synthesis of the educational material optimized to the characteristics of the student.

17 Claims, 79 Drawing figures
Exemplary Claim Number: 10

Number of Drawing Sheets: 58

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

| Terms | Documents |
|---|---|
| L1 AND loop | 36 |

**Display Format:** REV Change Format

Previous Page Next Page Go to Doc#

# Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

**Search Results** - Record(s) 1 through 36 of 36 returned.

☐ 1. Document ID: US 6820258 B1

L2: Entry 1 of 36 File: USPT Nov 16, 2004

US-PAT-NO: 6820258
DOCUMENT-IDENTIFIER: US 6820258 B1

TITLE: System and method for dynamically optimizing executing activations

DATE-ISSUED: November 16, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Fink; Stephen J. | Yorktown Heights | NY | | |
| Wegman; Mark | Ossining | NY | | |

US-CL-CURRENT: 717/158

ABSTRACT:

A system and method for dynamically optimizing a code sequence of a program while executing in a computer system comprises: identifying one or more program yield points in an original code sequence at which a run-time representation of the original code sequence may be optimized in an executing program; generating a prologue of instructions for setting up program state associated with the original code sequence at a particular yield point; adding the prologue of instructions to an intermediate representation of the original code sequence code for generating a specialized code sequence; and, compiling the specialized code sequence with a compiler for generating a run-time representation of the specialized code sequence, the run-time representation being further optimized for execution on a target computer system.

20 Claims, 3 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 3

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

☑ 2. Document ID: US 6820250 B2

L2: Entry 2 of 36 File: USPT Nov 16, 2004

US-PAT-NO: 6820250

DOCUMENT-IDENTIFIER: US 6820250 B2

TITLE: Mechanism for software pipelining loop nests

DATE-ISSUED: November 16, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Muthukumar; Kalyan | Cupertino | CA | | |
| Doshi; Gautam B. | Sunnyvale | CA | | |

US-CL-CURRENT: 717/116; 717/150

ABSTRACT:

A method is provided for processing nested loops that include a modulo-scheduled inner loop within an outer loop. The nested loop is scheduled to execute the epilog stage of the inner loop for a given iteration of the outer loop with the prolog stage of the inner loop for the next iteration of the outer loop. For one embodiment of the invention, this is accomplished by initializing an epilog counter for the inner loop to a value that bypasses draining the software pipeline. This causes the processor to exit the inner loop before it begins draining the inner loop pipeline. The inner loop pipeline is drained during the next iteration of the outer loop, while the inner loop pipeline fills for the next iteration of the outer loop.

29 Claims, 11 Drawing figures
Exemplary Claim Number: 9
Number of Drawing Sheets: 10



---

☐ 3. Document ID: US 6810519 B1

L2: Entry 3 of 36 File: USPT Oct 26, 2004

US-PAT-NO: 6810519
DOCUMENT-IDENTIFIER: US 6810519 B1

TITLE: Achieving tight binding for dynamically loaded software modules via intermodule copying

DATE-ISSUED: October 26, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Hicks; Daniel Rodman | Byron | MN | | |

US-CL-CURRENT: 717/166

ABSTRACT:

A computer system, a computer product and a method in which static storage within an environment comprising a plurality of compilation modules is managed such that compiled cloned copies of called externally resolved (with respect to a compilation unit) items are preferentially executed in favor of the corresponding externally resolved item based on a favorable comparison of version information of version information prior to execution. In one embodiment, JAVA.RTM. programming language methods are processed within the context a modified framework.

7 Claims, 23 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 15



---

☐ 4. Document ID: US 6785882 B1

L2: Entry 4 of 36 File: USPT Aug 31, 2004

US-PAT-NO: 6785882
DOCUMENT-IDENTIFIER: US 6785882 B1

TITLE: Process-driven tool interface for an object management system

DATE-ISSUED: August 31, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Goiffon; David A. | Shoreview | MN | | |
| Hartmann; Gerald E. | Minneapolis | MN | | |
| Johnson; David R. | Oakdale | MN | | |

US-CL-CURRENT: 717/120; 707/200

ABSTRACT:

A process-driven object management system for managing data and code modules is disclosed. The object management system includes a repository that stores objects, wherein ones of the objects referred to as "Asset elements" each describe a respective code or data module. The object management system includes a set of scripted tools for performing renovation, transformation, and code development tasks on the code and data modules. According to one aspect of the invention, the tool invocation constructs are stored as objects in the repository, such that some of the same object management tools and automated repository interface functions used to manage the Asset element can also be used to manage and veiw the tool objects.

46 Claims, 20 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 17

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

☐ 5. Document ID: US 6782374 B2

L2: Entry 5 of 36 File: USPT Aug 24, 2004

US-PAT-NO: 6782374
DOCUMENT-IDENTIFIER: US 6782374 B2

TITLE: System, method and article of manufacturing for a runtime program analysis tool for a simulation engine

DATE-ISSUED: August 24, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Nichols; Mark Stewart | Downers Grove | IL | | |

US-CL-CURRENT: 706/45; 706/15

ABSTRACT:

A system is disclosed that provides a goal based learning system utilizing a rule based expert training system to provide a cognitive educational experience. The system provides the user with a simulated environment that presents a business opportunity to understand and solve optimally. Mistakes are noted and remedial educational material presented dynamically to build the necessary skills that a user requires for success in the business endeavor. The system utilizes an artificial intelligence engine driving individualized and dynamic feedback with synchronized video and graphics used to simulate real-world environment and interactions. Multiple "correct" answers are integrated into the learning system to allow individualized learning experiences in which navigation through the system is at a pace controlled by the learner. A robust business model provides support for realistic activities and allows a user to experience real world consequences for their actions and decisions and entails realtime decision-making and synthesis of the educational material. The system includes tools for analysis and display of a presentation as it is presented.

19 Claims, 79 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 58

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

☐ 6. Document ID: US 6772415 B1

L2: Entry 6 of 36 File: USPT Aug 3, 2004

US-PAT-NO: 6772415
DOCUMENT-IDENTIFIER: US 6772415 B1

TITLE: Loop optimization with mapping code on an architecture

DATE-ISSUED: August 3, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Danckaert; Koen | Asse | | | BE |
| Catthoor; Francky | Temse | | | BE |

US-CL-CURRENT: 717/161; 712/241

ABSTRACT:

A loop transformation step, to be performed on code and improving data transfer and storage, while executing said transformed code on a parallel processor, is disclosed. Improval of the data locality and regularity of the algorithm, described by said code, is aimed at. Said loop transformation step works globally and is feasible for realistic code sizes.

47 Claims, 45 Drawing figures
Exemplary Claim Number: 47
Number of Drawing Sheets: 28



---

☐ 7. Document ID: US 6769113 B1

L2: Entry 7 of 36 File: USPT Jul 27, 2004

US-PAT-NO: 6769113
DOCUMENT-IDENTIFIER: US 6769113 B1

TITLE: Enterprise process models and enterprise application for information technologies

DATE-ISSUED: July 27, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Bloom; Bard | Yorktown Heights | NY | | |
| Reimer; Darrell | Ossining | NY | | |
| Simmonds; Ian D. | Dobbs Ferry | NY | | |
| Wegman; Mark N. | Ossining | NY | | |

US-CL-CURRENT: 717/103; 705/8

ABSTRACT:

An enterprise process model that comprises a plurality of actors, actions performed by said actors, objects acted upon by said actions, and roles. Data characterizing the view of the attributes of at least one of the objects of the model is generated by associating a plurality of situation/role pairs with the at least one object; and then, for each particular situation/role pair, defining a view definition for subsequent use. In another aspect of the present invention, an enterprise application for use in an information system that comprises diverse software services and hardware platforms is generated by providing a model of the enterprise

process, wherein the model comprises a plurality of actors, actions performed by said actors, objects acted upon by said actions, and roles. A configuration defining software services and hardware platforms that support the model is generated. Finally, program fragments that support the model are generated.

12 Claims, 9 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 9



---

☐ 8. Document ID: US 6763327 B1

L2: Entry 8 of 36 File: USPT Jul 13, 2004

US-PAT-NO: 6763327
DOCUMENT-IDENTIFIER: US 6763327 B1

TITLE: Abstraction of configurable processor functionality for operating systems portability

DATE-ISSUED: July 13, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Songer; Christopher Mark | Mountain View | CA | | |
| Konas; Pavlos | Mountain View | CA | | |
| Gauthier; Marc E. | Sunnyvale | CA | | |
| Chea; Kevin C. | San Francisco | CA | | |

US-CL-CURRENT: 703/21; 703/20, 703/22, 703/23, 703/27, 716/1, 716/17, 716/18, 716/2, 717/107, 717/138, 717/151, 717/163 , 718/1, 719/310, 719/319, 719/321, 719/324, 719/328

ABSTRACT:

A hardware abstraction layer operates as a system architectural layer between a real-time operating system and an underlying configurable processor. The hardware abstraction layer provides an abstraction of processor-specific functionality to the operating system. In particular, it abstracts configurable processor features visible to the operating system to provide a uniform, standardized interface between the operating system and the configurable processor on which it runs. Thus, an operating system running on top of the hardware abstraction layer will work on all configurations of the processor which differ from one another only in the configuration parameters covered by the hardware abstraction layer. The hardware abstraction layer may be generated using the same information that is used to describe the features being configured in the configurable processor. Automatic generation of the HAL greatly eases use of the HAL and the configurable processor, since the user is not required to manually write the HAL or adapt an existing one based on the processor configuration parameters.

16 Claims, 3 Drawing figures
Exemplary Claim Number: 1

Number of Drawing Sheets: 2



---

☐ 9. Document ID: US 6760903 B1

L2: Entry 9 of 36 File: USPT Jul 6, 2004

US-PAT-NO: 6760903
DOCUMENT-IDENTIFIER: US 6760903 B1

TITLE: Coordinated application monitoring in a distributed computing environment

DATE-ISSUED: July 6, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Morshed; Farokh | Amherst | NH | | |
| Meagher; Robert | Milford | NM | | |

US-CL-CURRENT: 717/130

ABSTRACT:

Techniques for gathering execution information about an application, such as a distributed application, are described. Key communication points in cross execution context calls, such as remote procedure calls, are determined and control is transferred to interception routines to insert and extract execution information. Outgoing remote procedure calls are intercepted on a client that inserts call origin information into the request sent to a server system. The server system intercepts and extracts the call origin information and additionally inserts other information in a response sent to the client system upon completion of a remote procedure call. In turn, the client system intercepts the response and extracts other performance information. On each client and server system, information is gathered by a reader and forwarded to a local collector. Program execution data may be collected and correlated for coordinated application monitoring.

50 Claims, 82 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 77

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

---

☐ 10. Document ID: US 6757888 B1

L2: Entry 10 of 36 File: USPT Jun 29, 2004

US-PAT-NO: 6757888
DOCUMENT-IDENTIFIER: US 6757888 B1

TITLE: Method and apparatus for manipulating data during automated data processing

DATE-ISSUED: June 29, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Knutson; Loren G. | Allen | TX | | |
| Zurawski; John C. | Allen | TX | | |
| Simone, Jr.; Kenneth D. | Murphy | TX | | |
| Gharbia; Nezar A. | Richardson | TX | | |

US-CL-CURRENT: 717/109; 345/619, 382/276, 715/515, 715/523, 717/106

ABSTRACT:

A number of items of data from a data source (12) can be processed and then deposited in at least one data destination (16, 17). The data can be image data, text data, numeric data or some other type of data, or a combination of these types of data. Processing of the data is controlled by a project definition (14, 71, 101) which includes a plurality of modules selected from a variety of available modules (Tables 1-4). The modules have input and output ports that are interrelated by binding information. Some of the modules are capable of taking an item of data and splitting it into two or more component parts. Other modules are capable of taking separate items of data and combining them.

14 Claims, 19 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 10



---

☐ 11. Document ID: US 6754885 B1

L2: Entry 11 of 36 File: USPT Jun 22, 2004

US-PAT-NO: 6754885
DOCUMENT-IDENTIFIER: US 6754885 B1

TITLE: Methods and apparatus for controlling object appearance in a process control configuration system

DATE-ISSUED: June 22, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Dardinski; Steven | Westford | MA | | |
| Eldridge; Keith | North Easton | MA | | |
| Hall; Robert | South Easton | MA | | |
| Johnson; Mark | North Attleboro | MA | | |
| MacKay; Brian | Coppell | TX | | |
| Meskonis; Paul | Norwood | MA | | |
| Volk; Scott | North Easton | MA | | |

US-CL-CURRENT: 717/113

ABSTRACT:

The invention provides improved apparatus for configuring process, environmental, industrial and other control systems. Such apparatus employs "appearance" objects (or other data and/or programming constructs) defining the appearance of configurable system components in graphical editors or other views in which the components may be depicted. "Placeholder" objects (or other constructs) persist the location, size, color, or other aspects of appearance defined by an appearance object for a configurable component in views in which it is actually depicted. By way of example, a process control configuration apparatus according to this aspect of the invention uses "configurable" objects to define blocks, loops and other components of a process control system. Appearance objects provide (or reference) icons or representations indicating how the configurable objects are to be depicted, e.g., in a configuration editor. Placeholder objects are created for each configurable object that is placed in a configuration using that editor. The placeholder objects identify the sizes, locations, colors, etc., of the icons used in the editor to represent the configurable objects.

70 Claims, 121 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 75



---

☐ 12. Document ID: US 6748585 B2

L2: Entry 12 of 36 File: USPT Jun 8, 2004

US-PAT-NO: 6748585
DOCUMENT-IDENTIFIER: US 6748585 B2

TITLE: Computer programming language pronouns

DATE-ISSUED: June 8, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Proebsting; Todd A. | Redmond | WA | | |
| Zorn; Benjamin G. | Woodinville | WA | | |

US-CL-CURRENT: 717/136; 717/114, 717/116, 717/140, 717/141, 717/142, 717/143

ABSTRACT:

Programming language constructs called pronouns and referents, and a method, system, and apparatus for translating computer source code that contains the pronouns and referents. A referent is any semantic or syntactic construct in the source code (e.g., a statement, a portion of a statement, an expression, or a value) to which a pronoun refers. A pronoun is a programming-language defined source-code symbol or a sequence of symbols that refers to the referent. As a result, pronouns eliminate the need to define new names or macros for repeated program segments. When a translator encounters the pronoun in the source code, the

translator searches the source code for the referent and substitutes the referent for the pronoun. Thus, by using pronouns and referents, the programmer can write programs faster and easier and eliminate program redundancy without losing readability.

31 Claims, 22 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 13



☐ 13. Document ID: US 6732353 B1

L2: Entry 13 of 36 File: USPT May 4, 2004

US-PAT-NO: 6732353
DOCUMENT-IDENTIFIER: US 6732353 B1

TITLE: Method and system for generating enterprise applications of a diversity of information technologies

DATE-ISSUED: May 4, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Bloom; Bard | Yorktown Heights | NY | | |
| Reimer; Darrell | Ossining | NY | | |
| Simmonds; Ian D. | Dobbs Ferry | NY | | |
| Wegman; Mark N. | Ossining | NY | | |

US-CL-CURRENT: 717/103; 705/8

ABSTRACT:

An enterprise process model that comprises a plurality of actors, actions performed by said actors, objects acted upon by said actions, and roles. Data characterizing the view of the attributes of at least one of the objects of the model is generated by associating a plurality of situation/role pairs with the at least one object; and then, for each particular situation/role pair, defining a view definition for subsequent use.

In another aspect of the present invention, an enterprise application for use in an information system that comprises diverse software services and hardware platforms is generated by providing a model of the enterprise process, wherein the model comprises a plurality of actors, actions performed by said actors, objects acted upon by said actions, and roles. A configuration defining software services and hardware platforms that support the model is generated. Finally, program fragments that support the model are generated.

40 Claims, 9 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 9

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

☐ 14. Document ID: US 6725452 B1

L2: Entry 14 of 36 File: USPT Apr 20, 2004

US-PAT-NO: 6725452
DOCUMENT-IDENTIFIER: US 6725452 B1

TITLE: Method for resolving dependency conflicts among multiple operative entities within a computing environment

DATE-ISSUED: April 20, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Te'eni; Moddy | Tel Aviv | | | IL |
| Shufer; Ilan | Bet Dagan | | | IL |

US-CL-CURRENT: 717/168

ABSTRACT:

A computer-based method sets up automatically a computer system configuration in association with a computer system upgrade process. In accordance with the system configuration, the upgrade process installs a required set of operative elements in order to provide for the reliable operation of the computer system. The operatively correct combination of hardware, system and application software components is accomplished by inter-component dependency checking and conflict resolving procedures utilizing specific inter-component dependency rules tables and component definition and inter-component relationship definition databases.

17 Claims, 10 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 10

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

☐ 15. Document ID: US 6718533 B1

L2: Entry 15 of 36 File: USPT Apr 6, 2004

US-PAT-NO: 6718533
DOCUMENT-IDENTIFIER: US 6718533 B1

TITLE: Method for building a real-time control system with mode and logical rate

DATE-ISSUED: April 6, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Schneider; Stanley A. | Sunnyvale | CA | | |
| Chen; Vincent W. | San Jose | CA | | |
| Pardo-Castellote; Gerardo | Palo Alto | CA | | |
| Wang; Howard H. | Sunnyvale | CA | | |
| Joshi; Rajive | Sunnyvale | CA | | |

US-CL-CURRENT: 717/100; 700/97

ABSTRACT:

A development tool combines advantages of a simulation tool with an object-oriented-modeling tool, including a real-time mathematical matrix library and an object model. The tool is applicable to any intelligent control system. A composite object group (COG) contains both sampled-data and event-driven capabilities. Mapping the computing resources of a computing device throughout the hierarchy of a control system is provided for modes, executable programs and threads of a processor. Within a given processor, each component of a control system may be assigned to individual execution threads within that processor simply by naming a thread.

23 Claims, 34 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 24



---

☑ 16. Document ID: US 6708331 B1

L2: Entry 16 of 36 File: USPT Mar 16, 2004

US-PAT-NO: 6708331
DOCUMENT-IDENTIFIER: US 6708331 B1

TITLE: Method for automatic parallelization of software

DATE-ISSUED: March 16, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Schwartz; Leon | Zichron Yaakov, 30900 | | | IL |

US-CL-CURRENT: 717/160

ABSTRACT:

The invention provides a scalable, automated, network friendly method for building parallel applications from embarrassingly parallel serial programs. Briefly, the steps of an exemplary method in this invention are as follows: First, the application loops with significant potential parallelism are identified. Second, from the set of loops identified, those loops which can statically be shown to not be parallelizable are disqualified. Next, the program is transformed into a parallel form in which the remaining identified loops are optimistically

parallelized and packaged into per-iteration functions. Then, shared memory in the functions is relocated to a shared memory section available to all computers and references to the shared memory in the source code are transformed into indirect accesses. Finally, the per-iteration functions are spawned on to multiple computers, at run-time, where each computer is given a range of iteration.

10 Claims, 12 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 6



---

☐ 17. Document ID: US 6684391 B1

L2: Entry 17 of 36 File: USPT Jan 27, 2004

US-PAT-NO: 6684391
DOCUMENT-IDENTIFIER: US 6684391 B1

TITLE: Method for operating a computer system, byte code verifier and computer system

DATE-ISSUED: January 27, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Stroetmann; Karl | Munich | | | DE |

US-CL-CURRENT: 717/148; 717/143

ABSTRACT:

The invention is directed to a method for operating a computer system, as well as to a byte code verifier and to a computer system. The inventive method checks whether a computer program loaded onto a computer system exercises an illegal access to a variable, i.e. whether the variable is initialized before it is read. This test ensues before the execution of the program, so that such a test no longer need be performed upon execution of the program. The inventive method requires little memory space in the testing of the program and nonetheless carries out a complete test. Since the testing ensues before the execution of the program, the program execution itself is considerably speeded up since no further test is thereby required.

11 Claims, 5 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4



---

☐ 18. Document ID: US 6681383 B1

L2: Entry 18 of 36 File: USPT Jan 20, 2004

US-PAT-NO: 6681383
DOCUMENT-IDENTIFIER: US 6681383 B1

TITLE: Automatic software production system

DATE-ISSUED: January 20, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Pastor; Oscar | Valencia | | | ES |
| Iborra; Jose | Denia Alicante | | | ES |

US-CL-CURRENT: 717/126

ABSTRACT:

An automated software production system is provided, in which system requirements are captured, converted into a formal specification, and validated for correctness and completeness. In addition, a translator is provided to automatically generate a complete, robust software application based on the validated formal specification, including user-interface code and error handling code.

43 Claims, 7 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 7



---

☐ 19. Document ID: US 6678886 B2

L2: Entry 19 of 36 File: USPT Jan 13, 2004

US-PAT-NO: 6678886
DOCUMENT-IDENTIFIER: US 6678886 B2

TITLE: Apparatus and method for generating optimization objects

DATE-ISSUED: January 13, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Kumon; Kouichi | Kawasaki | | | JP |

US-CL-CURRENT: 717/151; 717/135, 717/136, 717/140, 717/146, 717/154, 717/158, 717/159

ABSTRACT:

A system and method enable appropriately concentrating instruction strings or data pieces sporadically present in a plurality of regions over more than one

compilation unit and adjusting the front-and-rear relationship of executed instruction strings without changing the program compilation unit such as a file, subroutine, or function and also without creating a link processing program for batch processing of the system as a whole. Different section names are given to the executed instruction strings and the unexecuted instruction strings and the referenced data and the unreferenced data of an object program respectively. When an execution module is generated from the object program by linking, the sections having an executed section name and the sections having an unexecuted section name in a plurality of files may be aggregated respectively to divide the instructions into an execution portion and an unexecution portion.

25 Claims, 47 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 38



---

☐ 20. Document ID: US 6665866 B1

L2: Entry 20 of 36 File: USPT Dec 16, 2003

US-PAT-NO: 6665866
DOCUMENT-IDENTIFIER: US 6665866 B1
**** See image for Certificate of Correction ****

TITLE: Extensible compiler utilizing a plurality of question handlers

DATE-ISSUED: December 16, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Kwiatkowski; Paul | Seattle | WA | | |
| Richter; David | Seattle | WA | | |
| Aitken; William | Mercer Island | WA | | |
| Dickens; Brian | Bellevue | WA | | |
| Simonyi; Charles | Bellevue | WA | | |
| Paramasivam; Muthukrishnan | Seattle | WA | | |
| Eisner; Steve | Seattle | WA | | |
| Samaragdakis; Ioannis | Austin | TX | | |

US-CL-CURRENT: 717/159; 717/144, 717/158

ABSTRACT:

A compiler architecture uses a question and answer methodology between a reduction engine and nodes of a graph representing the program being compiled to provide for easy expandability of the compiler. By using the question and answer methodology, additional functionality can be added to the compiler by users, whether they be part of the original design team of the compiler or be a subsequent user.

51 Claims, 8 Drawing figures
Exemplary Claim Number: 1

Number of Drawing Sheets: 8

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

---

☐ 21. Document ID: US 6665865 B1

L2: Entry 21 of 36 File: USPT Dec 16, 2003

US-PAT-NO: 6665865
DOCUMENT-IDENTIFIER: US 6665865 B1

TITLE: Equivalence class based synchronization optimization

DATE-ISSUED: December 16, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Ruf; Erik | Redmond | WA | | |

US-CL-CURRENT: 717/157; 717/144, 717/151, 717/154, 717/156

ABSTRACT:

Synchronization optimization for statically compiled Java programs is performed in three phases: Thread closure analysis, Alias analysis, and Specialization and transformation. Thread closure analysis bounds the number of thread instances constructed at each thread allocation site, and determines the set of methods potentially executed by each thread instance. Alias analysis generates equivalence class representation based alias signatures for each method. These signatures describe the aliasing and synchronization behavior of each method. The specialization and transformation phase traverses a call graph in a top-down manner starting from the program entry point, and creates specialized copies of methods when they can be individually optimized. A synchronization operation is removed from the code whenever it can be proven that all objects reaching the operation at runtime can be synchronized by at most one thread instance.

30 Claims, 8 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 5

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

---

☐ 22. Document ID: US 6662357 B1

L2: Entry 22 of 36 File: USPT Dec 9, 2003

US-PAT-NO: 6662357
DOCUMENT-IDENTIFIER: US 6662357 B1
**** See image for Certificate of Correction ****

TITLE: Managing information in an integrated development architecture framework

DATE-ISSUED: December 9, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Bowman-Amuah; Michel K. | Colorado Springs | CO | | |

US-CL-CURRENT: 717/120

ABSTRACT:

A system, method, and article of manufacture are provided for managing information in a development architecture framework. Common information that is used by a plurality of components of a system is allowed to be accessed in a single, shared repository. Unique information that is unique to the components of the system is stored in corresponding designated folders. Media content communicated in the system is managed based on metadata thereof.

18 Claims, 14 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 14

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWMC | Draw. D

---

☐ 23. Document ID: US 6662355 B1

L2: Entry 23 of 36 File: USPT Dec 9, 2003

US-PAT-NO: 6662355
DOCUMENT-IDENTIFIER: US 6662355 B1

TITLE: Method and system for specifying and implementing automation of business processes

DATE-ISSUED: December 9, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Caswell; Nathan S. | Yorktown Heights | NY | | |
| Ciccolo; Arthur C. | Ridgefield | CT | | |
| Nigam; Anil | Stamford | CT | | |

US-CL-CURRENT: 717/103; 705/1

ABSTRACT:

A method s for specifying and implementing automation of business processes where the specification is independently manipulable by both the business process owner and technical implementers, and resulting technical elements can be tested for compliance with every detail in the specification. The method creates a single shared model suitable for understanding and execution in both the business and technical domains by focusing on the specification problem in the area of business automation. The solution to the specification problem lies in Information,

Function, Flow (IFF or IF.sup.2) factorization of business processes. Models of the business are constructed by way of the IF.sup.2 modeling methodology. This is a complete model which includes, by construction, external specifications of each task included in the business model. The modularization problem is solved by preserving the partitioning of the system created in the business model. The automation system implements concrete modules that uniquely and directly correspond to particular elements whose external specification is determined by the business model.

9 Claims, 5 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 5

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWMC | Draw D

---

☐ 24. Document ID: US 6654951 B1

L2: Entry 24 of 36 File: USPT Nov 25, 2003

US-PAT-NO: 6654951
DOCUMENT-IDENTIFIER: US 6654951 B1

TITLE: Removal of unreachable methods in object-oriented applications based on program interface analysis

DATE-ISSUED: November 25, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Bacon; David Francis | New York | NY | | |
| Laffra; Johannes C. | Yorktown Heights | NY | | |
| Sweeney; Peter Francis | Spring Valley | NY | | |
| Tip; Frank | Mount Kisco | NY | | |

US-CL-CURRENT: 717/154; 717/151

ABSTRACT:

The present invention analyzes an application A and computes a set reachable methods in A by determining the methods in A that may be called from another reachable method in A, or from within a class library L used by A without analyzing the classes in L. The invention may be used as an optimization to reduce application size by eliminating unreachable methods.

19 Claims, 5 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 5

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWMC | Draw D

---

☐ 25. Document ID: US 6634026 B1

L2: Entry 25 of 36 File: USPT Oct 14, 2003

US-PAT-NO: 6634026
DOCUMENT-IDENTIFIER: US 6634026 B1

TITLE: Method and apparatus for correcting common errors in multiple versions of a computer program

DATE-ISSUED: October 14, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Jones; Robert Dennis | Hartland | WI | | |

US-CL-CURRENT: 717/170

ABSTRACT:

The present invention relates to a repair program for multiple versions of computer programs that have a common error by using a pattern search and substitution technique. The invention includes identifying a common error in a main computer program, finding a common code section that contains the common error, and locating a segment of the common code section that is modifiable. The code section is then modified by optimizing the code to perform the same functionality and adding additional code to correct the error. A repair program is then written to search other versions of the main computer program and perform the modification step automatically without having to manipulate the source or machine code manually on the various versions of the software.

20 Claims, 3 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 3

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

---

☐ 26. Document ID: US 6631518 B1

L2: Entry 26 of 36 File: USPT Oct 7, 2003

US-PAT-NO: 6631518
DOCUMENT-IDENTIFIER: US 6631518 B1
** See image for Certificate of Correction **

TITLE: Generating and utilizing organized profile information

DATE-ISSUED: October 7, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Bortnikov; Vita | Haifa | | | IL |

| | | | |
|---|---|---|---|
| Lambert; David John | Rochester | MN | |
| Mendelson; Bilha | Haifa | | IL |
| Roediger; Robert Ralph | Rochester | MN | |
| Schmidt; William Jon | Rochester | MN | |
| Shavit-Lottem; Inbal | Kibbutz Bet-Oren | | IL |

US-CL-CURRENT: 717/158

ABSTRACT:

Disclosed is a system and method for a profiling system wherein profile data is stored in a separable hierarchical fashion such that profile data for each compiled procedure in a computer program can be readily identified and utilized. In particular, each source module has a corresponding profile data file and each procedure has a corresponding procedure profile area. The system and method also includes a mechanism for verifying the existence and validity of profile information, and a mechanism for handling invalid profile information.

29 Claims, 12 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 8



---

☐ 27. Document ID: US 6618856 B2

L2: Entry 27 of 36 File: USPT Sep 9, 2003

US-PAT-NO: 6618856
DOCUMENT-IDENTIFIER: US 6618856 B2

TITLE: Simulation method and apparatus for use in enterprise controls

DATE-ISSUED: September 9, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Coburn; James D. | Cleveland Heights | OH | | |
| Hoskins; Josiah C. | Austin | TX | | |
| Brooks; Ruven E. | Shorewood | WI | | |

US-CL-CURRENT: 717/135; 700/86, 703/15, 703/22

ABSTRACT:

A method used with a simulator and a controller, the controller running execution code to provide output signals which, when linked to resources, cause the resources to cycle through requested activities, the simulator receiving controller output signals and, in response thereto, generating motion pictures of resources as the resources cycle through requested activities, the simulator using data structures which model the resources to determine which motion pictures to generate, the method for generating execution code and data structures for use by the controller

and the simulator, respectively, and comprising the steps of, for each resource, encapsulating resource information including resource logic in a control assembly (CA), instantiating at least one instance of at least one CA, compiling instantiated CA instance resource logic to generate execution code, gleaning simulation information from the instantiated CA instances and using the gleaned simulation information to generate a simulation data structure for the resources corresponding to the instantiated CA instances.

26 Claims, 129 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 103



---

☐ 28. Document ID: US 6611956 B1

L2: Entry 28 of 36 File: USPT Aug 26, 2003

US-PAT-NO: 6611956
DOCUMENT-IDENTIFIER: US 6611956 B1

TITLE: Instruction string optimization with estimation of basic block dependence relations where the first step is to remove self-dependent branching

DATE-ISSUED: August 26, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Ogawa; Hajime | Neyagawa | | | JP |
| Odani; Kensuke | Kyoto | | | JP |

US-CL-CURRENT: 717/152

ABSTRACT:

An instruction string optimization apparatus is provided which estimates the size of a constant to be resolved as an address difference before linking instructions. The apparatus comprises code dividing means (202) for dividing a serial assembler code (201) into basic blocks, size dependence relation generation means (204) for analyzing size dependence relations among the sizes of the instruction string between basic blocks, estimation order determining means (206) for determining the order of basic blocks in which the size of a constant to be resolved as an address difference is determined and size determining means (208) for determining the size of the constant in each basic block according to the determined order, whereby the size of a constant to be resolved as an address difference can be estimated to be a value close to and not less than its actual size, the number of codes can be reduced, and the process speed by a linker can be improved.

11 Claims, 63 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 37

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

☐ 29. Document ID: US 6609249 B2

L2: Entry 29 of 36 File: USPT Aug 19, 2003

US-PAT-NO: 6609249
DOCUMENT-IDENTIFIER: US 6609249 B2

TITLE: Determining maximum number of live registers by recording relevant events of the execution of a computer program

DATE-ISSUED: August 19, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Kunz; Robert C. | Palo Alto | CA | | |
| Dahl; Peter J. | Cupertino | CA | | |

US-CL-CURRENT: 717/161

ABSTRACT:

The present invention is a method and apparatus for compiler optimization that determines the maximum number of live computer registers, or pressure point. The present invention improves the productivity of a software developer by reducing compilation time of a computer program. More particularly, the overhead required during compilation to search information to determine the maximum number of live registers is reduced. The present invention records the relevant events related to the execution of a computer program, as opposed to a comprehensive history of the read instructions and write instructions. Also, the present invention maintains information about the maximum number of live registers for any partition related to the execution of a computer program. The present invention may bound the required system resources required to determine the maximum number of live registers to the number of registers associated with the number of partitions.

12 Claims, 11 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 11

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

☐ 30. Document ID: US 6601236 B1

L2: Entry 30 of 36 File: USPT Jul 29, 2003

US-PAT-NO: 6601236
DOCUMENT-IDENTIFIER: US 6601236 B1
**** See image for Certificate of Correction ****

TITLE: Cross platform program installation on drives using drive object

DATE-ISSUED: July 29, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Curtis; Bryce Allen | Round Rock | TX | | |

US-CL-CURRENT: 717/177

ABSTRACT:

Disclosed is a set of tools or program instructions, an installation program, and a system that operates a drive in a platform independent manner. A drive object represents a single drive mounted by an operating system and contains fields providing information including drive name, block size, free space, type, format long file name support, and space needed by the files to be installed on that drive. During the installation process, each drive is processed to ensure that the drive has sufficient free space for the files to be installed.

30 Claims, 4 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4



---

☐ 31. Document ID: US 6578192 B1

L2: Entry 31 of 36 File: USPT Jun 10, 2003

US-PAT-NO: 6578192
DOCUMENT-IDENTIFIER: US 6578192 B1

TITLE: Method and system for supporting dynamic document content expressed in a component-level language

DATE-ISSUED: June 10, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Boehme; Richard F. | Kent Lakes | NY | | |
| Duftler; Matthew J. | Tarrytown | NY | | |
| Epstein; David A. | Ossining | NY | | |
| Kesselman; Joseph J. | Ossining | NY | | |
| Weerawarana; Sanjiva | Yorktown Heights | NY | | |

US-CL-CURRENT: 717/115

ABSTRACT:

This invention provides a computer-method for parsing by enabling scripts to be expressed in a language which is syntax-compatible with the document surrounding them. A document is loaded having script and non-script components. Script components are identified and delineated, and are then passed to an interpreter,

which returns an object corresponding to each script component. Then, script elements in the original document are replaced with the last returned object from the interpreter. If the returned object is a suitable Document Object Model (DOM) Node, it replaces the script element in the document structure. If the object is not a DOM node, the server invokes its string conversion method to obtain a textual representation, and replaces the script element with that text. After all BML markup block elements in the document have been processed, the altered document is delivered to the client.

14 Claims, 4 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWMC | Draw De

---

☑ 32. Document ID: US 6571385 B1

L2: Entry 32 of 36 File: USPT May 27, 2003

US-PAT-NO: 6571385
DOCUMENT-IDENTIFIER: US 6571385 B1
**** See image for Certificate of Correction ****

TITLE: Early exit transformations for software pipelining

DATE-ISSUED: May 27, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Muthukumar; Kalyan | Cupertino | CA | | |
| Chen; Dong-Yuan | Fremont | CA | | |
| Wu; Youfeng | Palo Alto | CA | | |
| Lavery; Daniel M. | Santa Clara | CA | | |

US-CL-CURRENT: 717/150; 712/219, 712/239

ABSTRACT:

The invention is directed to the transformation of software loops having early exit conditions, thereby allowing the loops to be more effectively converted to a single basic block for software pipelining. The invention assigns a predicate register for each early exit condition of the software loop. The predicate registers are set when the corresponding early exit condition is satisfied. In this manner, when the loop terminates the predicate registers can be examined to indicate which early exit conditions were satisfied. The invention produces loops having a lower recurrence II and resource II than conventional techniques.

18 Claims, 7 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 7

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWMC | Draw De

☐ 33. Document ID: US 6564179 B1

L2: Entry 33 of 36 File: USPT May 13, 2003

US-PAT-NO: 6564179
DOCUMENT-IDENTIFIER: US 6564179 B1

TITLE: DSP emulating a microcontroller

DATE-ISSUED: May 13, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Belhaj; Said O. | Coplay | PA | | |

US-CL-CURRENT: 703/26; 703/24, 703/25, 712/203, 712/209, 712/227, 712/28, 712/31, 712/34, 712/35, 712/36

ABSTRACT:

The present invention provides a processor device and technique having the capability of providing a two-processor solution with only one processor. In accordance with the principles of the present invention, a host processor is programmed in its native source and machine code language, and an emulated second processor is programmed in a different native source or machine code language particular to that emulated processor, to allow programming specialists in the different processors to develop common code for use on the same host processor. A multitasking operating system is included to allow time sharing operation between instructions from program code relating to the host processor (e.g., a DSP in the disclosed embodiment), and different program code relating to the emulated processor. The program code relating to the host processor (e.g., DSP) is written in program code which is native to the DSP, while the program code relating to the emulated processor (e.g., microcontroller) is written in program code which is native to the microcontroller. The SoftCore emulation module allows both DSP code and control code written for a microcontroller to execute independently on the same processor by multi-tasking resources in the faster, host processor (e.g., in the DSP), giving equal time slots of processor time to each processor (real and emulated).

4 Claims, 5 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 5

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWMC | Draw D

☐ 34. Document ID: US 6546362 B1

L2: Entry 34 of 36 File: USPT Apr 8, 2003

US-PAT-NO: 6546362
DOCUMENT-IDENTIFIER: US 6546362 B1

TITLE: Mold design system and recording medium

DATE-ISSUED: April 8, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Guo; Fu | Kawasaki | | | JP |
| Yoshikawa; Tadakatsu | Kawasaki | | | JP |

US-CL-CURRENT: 703/1; 703/2, 703/6, 703/7

ABSTRACT:

To quickly determine a parting line, a mold design system first obtains the orientations of faces constituting an article to be produced using a mold, and then classifies the faces according to their orientations. A boundary between faces that are classified into different sets is determined as a parting line. Thus, the parting line is automatically determined, permitting efficient mold design.

6 Claims, 22 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 22

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

---

☐ 35. Document ID: US 6539539 B1

L2: Entry 35 of 36 File: USPT Mar 25, 2003

US-PAT-NO: 6539539
DOCUMENT-IDENTIFIER: US 6539539 B1

TITLE: Active probes for ensuring software package compatibility

DATE-ISSUED: March 25, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Larsen; Mark S. | Hoffman Estates | IL | | |
| Liesen; Christopher D. | Naperville | IL | | |
| Zorn; Alan R. | Naperville | IL | | |

US-CL-CURRENT: 717/124; 379/9.01, 714/40, 717/121

ABSTRACT:

In a distributed computer program, active software probes in the form of small functions built into an application are invoked by another application. When invoked, an active probe provides a positive response if the service being requested is available from the probed package. If the service is not available, the probe will fail alerting the software package installer that there is a problem. The active probes thus perform a functionality check for the software

package, not a check merely based on the package's release number. Because the probe is active, it is capable of checking for subtending capabilities that the calling application may not realize are necessary for this service. This ensures full coverage of the test while hiding some implementation details from the calling application. By invoking these active probes upon installation of a new package, the installer can know immediately if the package has all services necessary to run correctly, thus reducing the possibility of a bad software package installation. The active software probes are thus capable of verifying software package compatibility at the time of program compilation as well as for program upgrades.

6 Claims, 6 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4



---

☐ 36. Document ID: US 6535861 B1

L2: Entry 36 of 36 File: USPT Mar 18, 2003

US-PAT-NO: 6535861
DOCUMENT-IDENTIFIER: US 6535861 B1

TITLE: Goal based educational system with support for dynamic characteristics tuning using a spread sheet object

DATE-ISSUED: March 18, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| O'Connor; Martha Torrey | Verona | NJ | | |
| Rosenfeld; Eren Tolga | New York | NY | | |

US-CL-CURRENT: 706/11; 434/107, 434/322, 434/327, 706/45, 706/46, 706/47, 706/60

ABSTRACT:

A system is disclosed that provides a goal based learning system utilizing a rule based expert training system to provide a cognitive educational experience. The system provides the user with a simulated environment that presents a business opportunity to understand and solve optimally. Mistakes are noted and remedial educational material presented dynamically to build the necessary skills that a user requires for success in the business endeavor. The system utilizes an artificial intelligence engine driving individualized and dynamic feedback with synchronized video and graphics used to simulate real-world environment and interactions. Multiple "correct" answers are integrated into the learning system to allow individualized learning experiences in which navigation through the system is at a pace controlled by the learner. A robust business model provides support for realistic activities and allows a student to experience real world consequences for their actions and decisions and entails realtime decision-making and synthesis of the educational material optimized to the characteristics of the student.

17 Claims, 79 Drawing figures
Exemplary Claim Number: 10

Number of Drawing Sheets: 58

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

| Terms | Documents |
|---|---|
| L1 AND loop | 36 |

**Display Format:** REV Change Format

Previous Page Next Page Go to Doc#

# Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

**Search Results** - Record(s) 1 through 10 of 10 returned.

☐ 1. Document ID: US 6718535 B1

L3: Entry 1 of 10 File: USPT Apr 6, 2004

US-PAT-NO: 6718535
DOCUMENT-IDENTIFIER: US 6718535 B1
** See image for Certificate of Correction **

TITLE: System, method and article of manufacture for an activity framework design in an e-commerce based environment

DATE-ISSUED: April 6, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Underwood; Roy Aaron | Long Grove | IL | | |

US-CL-CURRENT: 717/101; 717/120

ABSTRACT:

A system and method are provided for providing an activity framework. First, a plurality of sub-activities are created which each include sub-activity logic adapted to generate an output based on an input received from a user upon execution. Second, a plurality of activities are defined which each execute the sub-activities in a unique manner upon being selected for accomplishing a goal associated with the activity. Selection of one of the activities is allowed by receiving user indicia. An interface is depicted for allowing receipt of the input and display of the output during execution of the sub-activities associated with the selected activity.

24 Claims, 179 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 111

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

☐ 2. Document ID: US 6704873 B1

L3: Entry 2 of 10 File: USPT Mar 9, 2004

US-PAT-NO: 6704873
DOCUMENT-IDENTIFIER: US 6704873 B1

TITLE: Secure gateway interconnection in an e-commerce based environment

DATE-ISSUED: March 9, 2004

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Underwood; Roy Aaron | Long Grove | IL | | |

US-CL-CURRENT: 713/201; 709/223, 709/249

ABSTRACT:

A system and method of providing a global internetworking gateway architecture in an e-commerce environment are provided. A plurality of gateways each situated in a distinct geographic location are coupled to an internet. A wide area network, separate from the internet, is coupled to each of the gateways for providing communication between the wide area network and the internet. Coupled to the wide area network is a central database for providing a central storage for data used in e-commerce carried out over the internet. In one embodiment, at least one of the gateways includes at least one screening router coupled to the internet service provider, at least one firewall connected to the screening router, and a choker router coupled between the wide area network and the firewall.

16 Claims, 179 Drawing figures
Exemplary Claim Number: 5
Number of Drawing Sheets: 111



---

☐ 3. Document ID: US 6633878 B1

L3: Entry 3 of 10 File: USPT Oct 14, 2003

US-PAT-NO: 6633878
DOCUMENT-IDENTIFIER: US 6633878 B1

TITLE: Initializing an ecommerce database framework

DATE-ISSUED: October 14, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Underwood; Roy Aaron | Long Grove | IL | | |

US-CL-CURRENT: 707/100; 707/1, 707/102, 707/205

ABSTRACT:

A system, method and article of manufacture are provided for initializing a database used with an issue tracker. The issue tracker receives information relating to a plurality of issues from a plurality of users, displays the information relating to the issues, and allows the browsing of the information relating to each of the issues. To initialize the database, the information

relating to the issues is stored in a first database. A second database is also provided that stores tables including: a plurality of user interfaces; and/or application logic for accessing the information in the first database. The tables of the second database are reconfigured upon migrating the first database from a first folder to a second folder.

15 Claims, 179 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 111

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

---

☐ 4. Document ID: US 6609128 B1

L3: Entry 4 of 10 File: USPT Aug 19, 2003

US-PAT-NO: 6609128
DOCUMENT-IDENTIFIER: US 6609128 B1

TITLE: Codes table framework design in an E-commerce architecture

DATE-ISSUED: August 19, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Underwood; Roy Aaron | Long Grove | IL | | |

US-CL-CURRENT: 707/10; 707/200

ABSTRACT:

A system, method and article of manufacture are provided for maintaining application consistency. First, a table of codes and associated text phrases are provided. Such table of codes is stored on a local storage medium within an e-commerce computer architecture. Next, the table of codes is accessed on the local storage medium within the e-commerce computer architecture. One of the text phrases is subsequently retrieved by selecting a corresponding one of the codes of the table. During operation, modification of the text phrases associated with each of the codes of the table is permitted. A plurality of services are executed, including retrieving a single one of the text phrases, retrieving all of the text phrases in response to a single command, updating a single code and text phrase combination, updating all of the code and text phrase combinations, naming the table, adding a new code and text phrase combination, removing one of the code and text phrase combination, and adding another table.

15 Claims, 179 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 111

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

---

☐ 5. Document ID: US 6601233 B1

L3: Entry 5 of 10 File: USPT Jul 29, 2003

US-PAT-NO: 6601233
DOCUMENT-IDENTIFIER: US 6601233 B1

TITLE: Business components framework

DATE-ISSUED: July 29, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Underwood; Roy Aaron | Long Grove | IL | | |

US-CL-CURRENT: 717/102; 717/100, 717/101, 717/103, 717/104, 717/106, 717/107

ABSTRACT:

A method of generating software based on business components. A plurality of logical business components in a business are first defined with each business component having a plurality of capabilities. Next, functional interrelationships are identified between the logical business components. Code modules are then generated to carry out the capabilities of the logical business components and the functional interrelationships between the logical business components, wherein the code modules represent a transformation of the logical business components to their physical implementation, while ensuring the capabilities that are carried out by each code module are essentially unique to the logical business component associated with the code module. Next, the functional aspects of the code modules and the functional relationships of the code modules are tested. The code modules are then subsequently deployed in an e-commerce environment.

18 Claims, 177 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 111

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

☐ 6. Document ID: US 6523027 B1

L3: Entry 6 of 10 File: USPT Feb 18, 2003

US-PAT-NO: 6523027
DOCUMENT-IDENTIFIER: US 6523027 B1
** See image for Certificate of Correction **

TITLE: Interfacing servers in a Java based e-commerce architecture

DATE-ISSUED: February 18, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Underwood; Roy Aaron | Long Grove | IL | | |

US-CL-CURRENT: 707/4; 707/10, 707/100

ABSTRACT:

A system, method and article of manufacture are provided for providing an interface between a first server and a second server with a proxy component situated therebetween. Initially, a request for a business object is identified by an application on the first server. The first server is then connected to the second server. Next, selection criteria from the first server is transmitted to the second server. In response to the selection criteria, the first server receives a first recordset and a second recordset from the second server. Business data is included in the first recordset and result codes are included in the second recordset. The first and second recordsets are mapped to the business object and the business object is sent to the application on the first server.

18 Claims, 179 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 111



---

☐ 7. Document ID: US 6513146 B1

L3: Entry 7 of 10 File: USPT Jan 28, 2003

US-PAT-NO: 6513146
DOCUMENT-IDENTIFIER: US 6513146 B1

TITLE: Method of designing semiconductor integrated circuit device, method of analyzing power consumption of circuit and apparatus for analyzing power consumption

DATE-ISSUED: January 28, 2003

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Yonezawa; Tomonori | Fukuoka | | | JP |
| Sasaki; Takayuki | Fukuoka | | | JP |
| Kondo; Takahiro | Fukuoka | | | JP |
| Otsuki; Hiroki | Fukuoka | | | JP |
| Nakamura; Tsuyoshi | Fukuoka | | | JP |

US-CL-CURRENT: 716/7; 716/2

ABSTRACT:

The processing quantity of each description part is estimated through a source code analysis of a system operation description language or through simulation, or power consumption of each function is estimated through an operation description analysis of functions. Predetermined threshold values are set with respect to the processing quantity and the power consumption of each description part or function, so as to determine S/W and H/W implementation, and then, S/W and H/W partitioning is carried out. Thereafter, it is determined whether or not the total processing quantity or

the total power consumption satisfies a desired design condition. Also, the S/W and H/W partitioning can be adjusted again in comprehensive consideration of the power consumption and the processing quantity, and the accuracy in the S/W and H/W partitioning can be improved by providing an instruction set simulator with a function to analyze power consumption. Moreover, an interface between S/W and H/W can be generated in the S/W and H/W partitioning so as to be automatically inserted into a S/W implemented part or a H/W implemented part.

22 Claims, 74 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 45



---

☑ 8. Document ID: US 6026240 A

L3: Entry 8 of 10 File: USPT Feb 15, 2000

US-PAT-NO: 6026240
DOCUMENT-IDENTIFIER: US 6026240 A
**** See image for Certificate of Correction ****

TITLE: Method and apparatus for optimizing program loops containing omega-invariant statements

DATE-ISSUED: February 15, 2000

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Subramanian; Krishna | Mountain View | CA | | |

US-CL-CURRENT: 717/161

ABSTRACT:

Apparatus, methods, and computer program products are disclosed for optimizing programs containing single basic block natural loops with a determinable number of iterations. The invention optimizes, for execution speed, such program loops containing statements that are initially variant, but stabilize and become invariant after some number of iterations of the loop. The invention optimizes the loop by unwinding iterations from the loop for which the statements are variant, and by hoisting the stabilized statement from subsequent iterations of the loop.

10 Claims, 11 Drawing figures
Exemplary Claim Number: 7
Number of Drawing Sheets: 9

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

---

☑ 9. Document ID: US 5930510 A

L3: Entry 9 of 10 File: USPT Jul 27, 1999

US-PAT-NO: 5930510
DOCUMENT-IDENTIFIER: US 5930510 A

TITLE: Method and apparatus for an improved code optimizer for pipelined computers

DATE-ISSUED: July 27, 1999

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Beylin; Boris | Palo Alto | CA | | |
| Subramanian; Krishna | Cupertino | CA | | |

US-CL-CURRENT: 717/161

ABSTRACT:

Apparatus, methods, systems and computer program products are disclosed to provide improved optimizations of single-basic-block-loops. These optimizations include improved scheduling of blocking instructions for pipelined computers and improved scheduling and allocation of resources (such as registers) that cannot be spilled to memory. Scheduling of blocking instructions is improved by pre-allocating space in the scheduling reservation table. Improved scheduling and allocation of non-spillable resources results from converting the resource constraint into a data dependency constraint.

24 Claims, 11 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 10



---

☐ 10. Document ID: US 5887174 A

L3: Entry 10 of 10 File: USPT Mar 23, 1999

US-PAT-NO: 5887174
DOCUMENT-IDENTIFIER: US 5887174 A

TITLE: System, method, and program product for instruction scheduling in the presence of hardware lookahead accomplished by the rescheduling of idle slots

DATE-ISSUED: March 23, 1999

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Simons; Barbara Bluestein | Palo Alto | CA | | |
| Sarkar; Vivek | Palo Alto | CA | | |

US-CL-CURRENT: 717/161; 712/216, 713/502, 717/156

ABSTRACT:

Instructions are scheduled for execution by a processor having a lookahead buffer by identifying an idle slot in a first instruction schedule of a first basic block of instructions, and by rescheduling the idle slot later in the first instruction schedule. The idle slot is rescheduled by determining if the first basic block of instructions may be rescheduled into a second instruction schedule in which the identified idle slot is scheduled later than in the first instruction schedule. The first basic block of instructions is rescheduled by determining a completion deadline of the first instruction schedule, decreasing the completion deadline, and determining the second instruction schedule based on the decreased completion deadline. Deadlines are determined by computing a rank of each node of a DAG corresponding to the first basic block of instructions; constructing an ordered list of the DAG nodes in nondecreasing rank order; and applying a greedy scheduling heuristic to the ordered list. An instruction in a second subsequent basic block of instructions may be rescheduled to execute in the rescheduled idle slot. This process may be repeated for each idle slot.

18 Claims, 15 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 10



| Terms | Documents |
|---|---|
| Loop And (do ADJ while) or (do-while) and pipelining | 10 |